

ECE 741/841

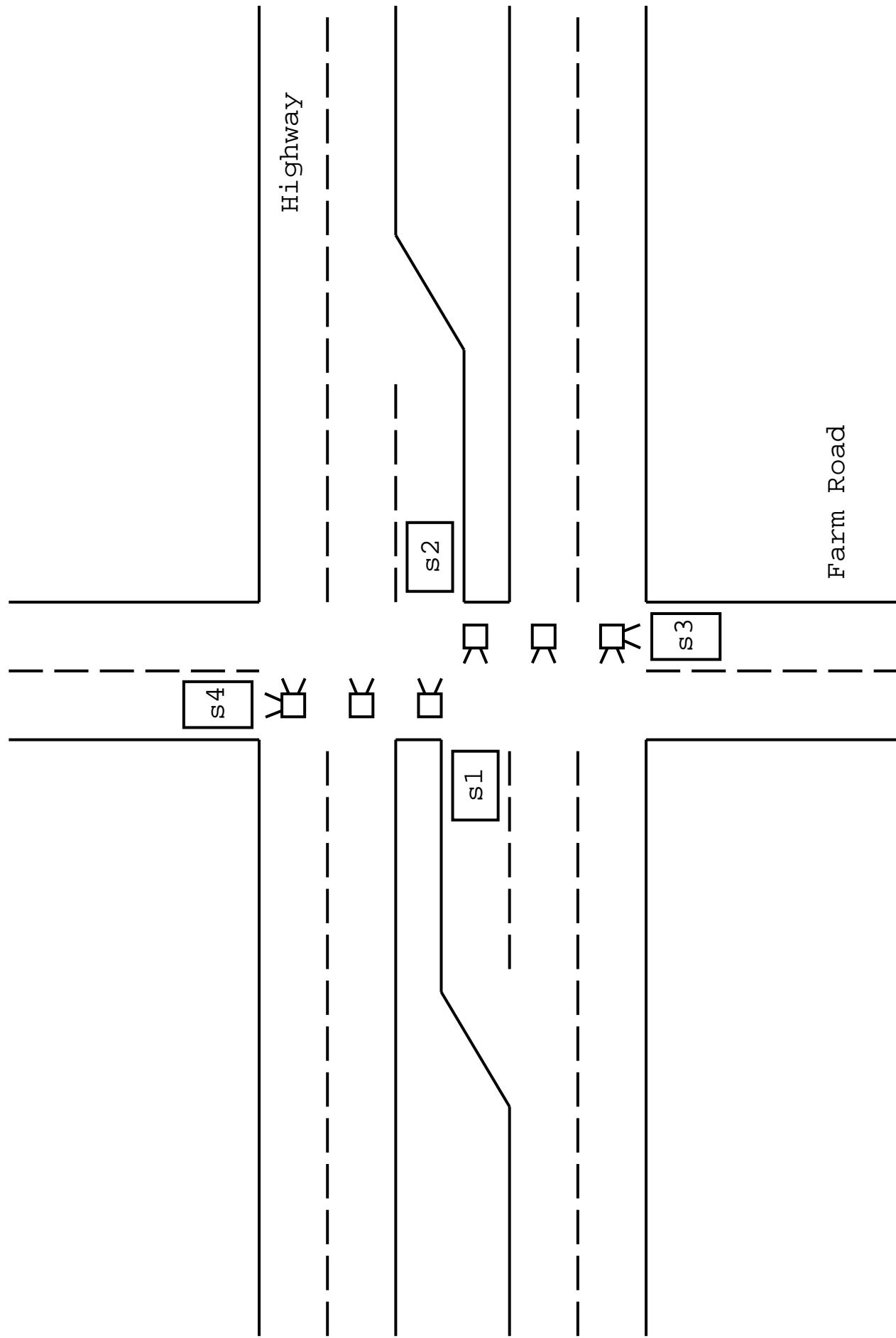
31 October 2002

Project Example 1, Traffic Light

Description

- Control for a traffic light in a 4 way intersection.
- Intersection has primary highway and secondary farm road.
- The highway is a four lane road and the farm road is a two lane road.
- There are dedicated left turn lanes on the highway road but not on the farm road.

Highway Intersection



Traffic Light Controller Requirements

- Highway traffic has priority over farm road; however, farm road traffic should not be stopped indefinitely.
- Traffic going straight on the highway has priority over traffic turning left; however, left turn traffic should be given turn arrow after a maximum wait.
- Farm road and left turn lanes have sensors that signal the controller of cars on the farm road and left turn lanes.
- In the absence of cars in the farm road, the highway traffic continues to have access to the intersection.
- If a sensor detects a car on the farm road, the controller should give access to the farm road with a delay of no more than $T_{farm_road_delay}$.

- The farm road light will remain green as long as the sensor detects cars on the farm road, but not longer than $T_{farm\text{-}road\text{-}green_max}$.
 - If a sensor detects a car on the left turn lane, the controller should give access to the left turn with a delay of no more than $T_{left\text{-}turn\text{-}delay}$.
 - The turn left light will remain green as long as the sensor detects cars on the left turn, but not longer than $T_{left\text{-}turn\text{-}green_max}$.
- The farm road light and the left turn light will cycle through yellow and red as soon as no cars are detected by the sensors.
- The highway light will be green a minimum of $T_{highway\text{-}green_min}$.

- The yellow light will have a minimum and maximum duration on the highway and farm roads;

$T_{yellow_highway_min}$, $T_{yellow_highway_max}$,

$T_{yellow_farm_road_min}$, $T_{yellow_farm_road_max}$.

- The controller will never give access to the highway and farm road simultaneously.
- The controller will never give access to left turn traffic and incoming traffic simultaneously.
- The controller will never give access to left turn and farm road traffic simultaneously.

System Properties

- Liveness
- Fairness
- Safety
- Correctness

Liveness

A liveness property says that the system will perform some (good) function.

In a state machine, the system will be able to transition from an initial state to a new state.

Fairness

A fairness property says that the system does not exclude any (good) function from being performed.

In a state machine, there is a possible path to any (good) state. All good states are reachable.

Safety

A safety property says that the system will not have any undesirable behavior.

Bad states are not reachable.

Correctness

The system will perform all functions in accordance to some notion of correctness.

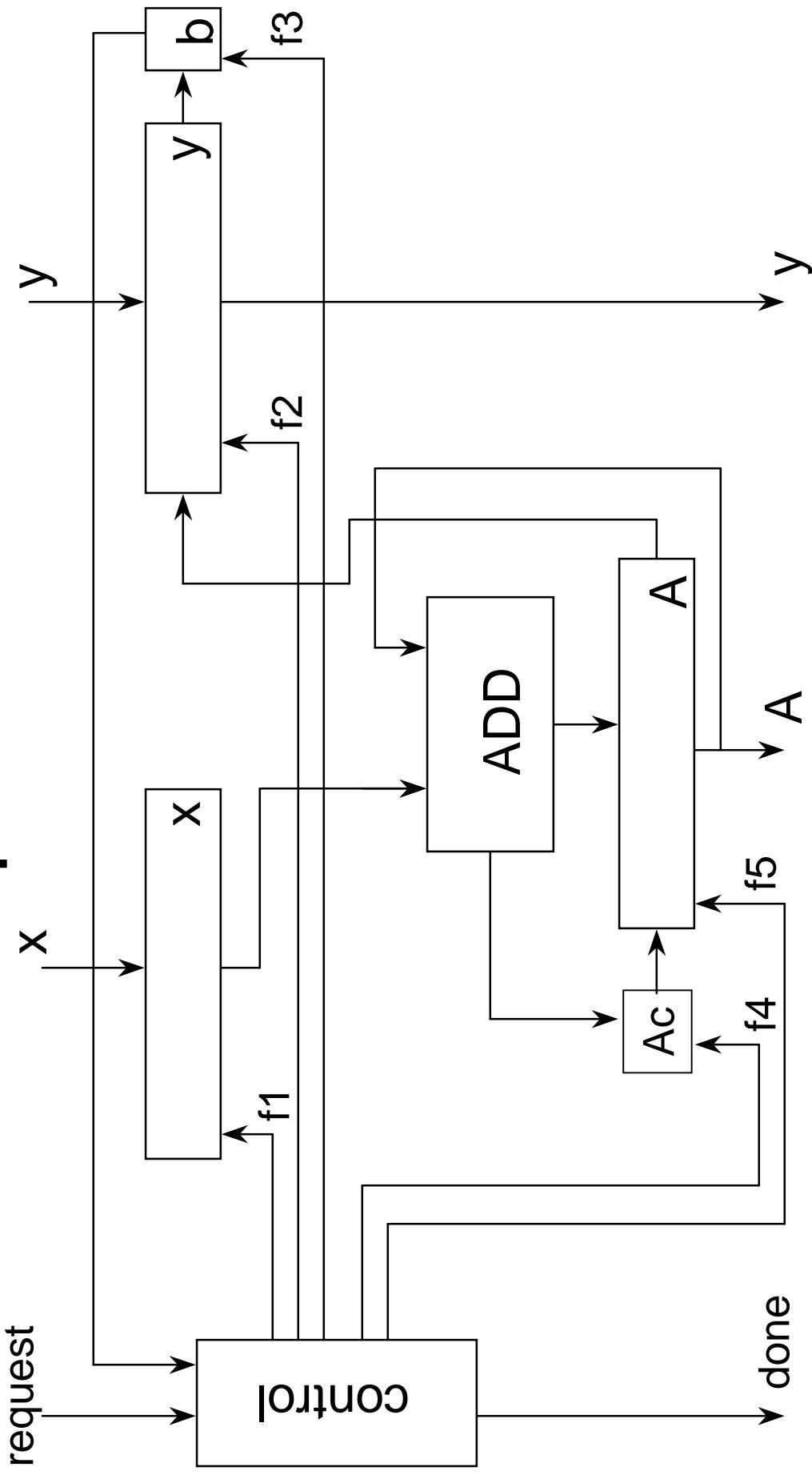
The functions performed by the system are the intended ones. The system meets a set of requirements.

The reachable states and sequences of states conform to a correctness criteria.

Project Example 2, Multiplier

- Micro-coded Multiplier
- Multiplication by shifting and repeated addition.

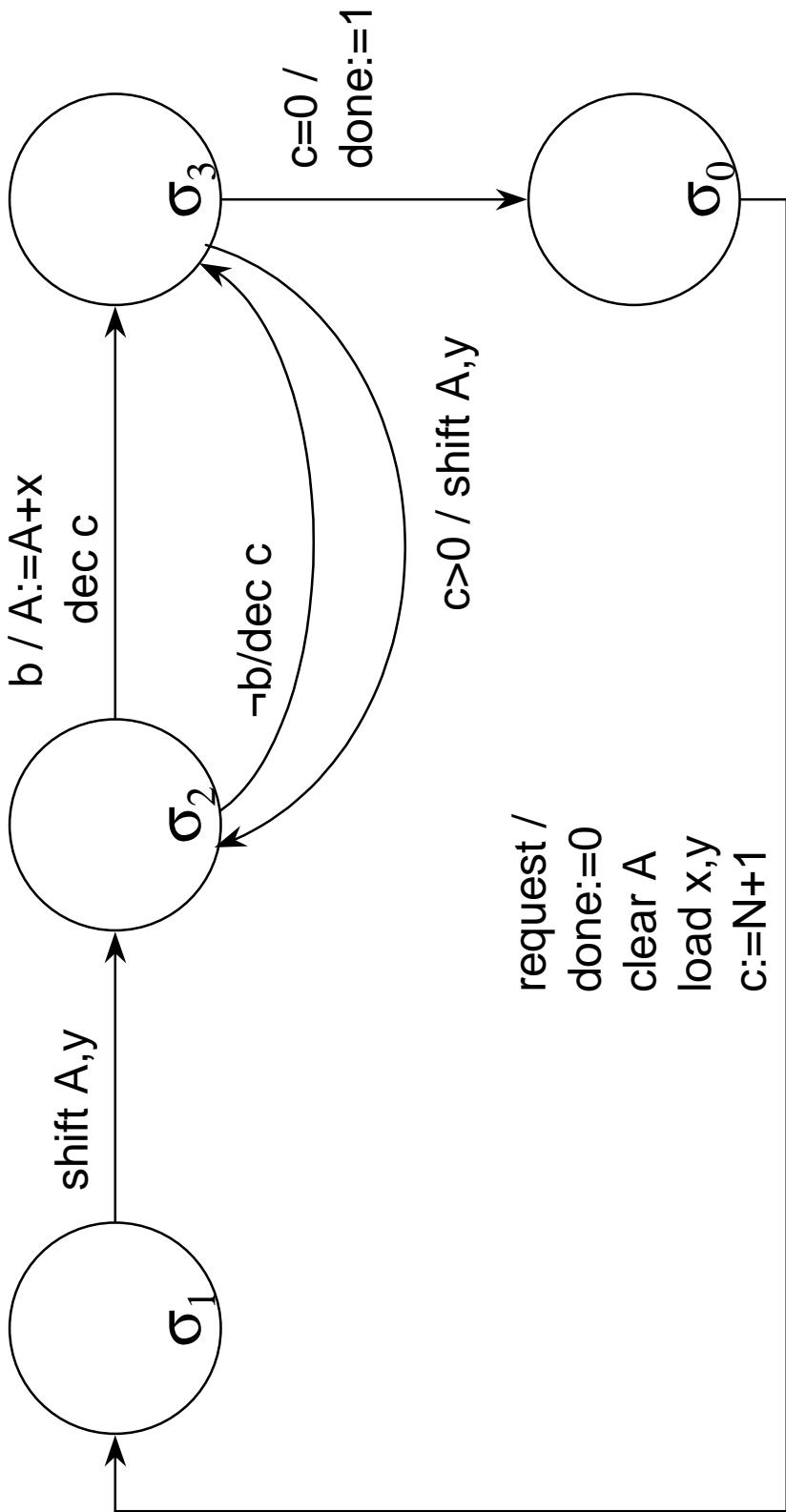
Multiplication Circuit



Multplier Loop

```
start
load y,x
clear A
set counter c := N+1
shift A,y (LSBA ->MSBy)
IF b=1
    A:=A+x
decrement counter
IF counter =0
    end
jump loop
```

Multiplication as State Machine



Registers' Control Line

N bit register

0 reset

1 unchanged

2 load

3 shift

One bit register

0 reset

1 unchanged

2 load/shift

Control Module

| | next | σ | f1 | f2 | f3 | f4 | f5 |
|---------------------------------|------------|----------|----|----|----|----|----|
| $\sigma_0, \neg \text{request}$ | σ_0 | 1 | 1 | 1 | 1 | 1 | 1 |
| $\sigma_0, \text{request}$ | σ_1 | 2 | 2 | 1 | 0 | 0 | 0 |
| σ_1 | σ_2 | 1 | 3 | 2 | 1 | 3 | 3 |
| $\sigma_2, b=1$ | σ_3 | 1 | 1 | 1 | 2 | 2 | 2 |
| $\sigma_2, b=0$ | σ_3 | 1 | 1 | 1 | 1 | 1 | 1 |
| $\sigma_3, c=0$ | σ_0 | 1 | 1 | 1 | 1 | 1 | 1 |
| $\sigma_3, c>0$ | σ_2 | 3 | 3 | 3 | 3 | 3 | 3 |

Control Module in PVS

```

control(sigma:below(4), c:upto(N), request:bool, b:bvec[1]: [below(4),
upto(N), bool, below(4)], below(4), below(3), below(4)] =
IF      sigma=1 THEN (2,c,false,1,3,2,1,3)
ELSIF   sigma=2 THEN
        IF b=fill[1](1) THEN (3,c-1,false,1,1,1,2,2)
        ELSE                  (3,c-1,false,1,1,1,1,1) ENDIF
ELSIF   sigma=3 THEN
        IF c=0      THEN (0,c,true,1,1,1,1,1)
        ELSE                  (2,c,false,1,3,2,1,3) ENDIF
ELSE %sigma=0
        IF request THEN (1,N,false,2,2,1,0,0)
        ELSE                  (0,c,true,1,1,1,1,1) ENDIF
ENDIF

```